

JC781 U.S. PTO
09/26/00

09-28-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

JC836 U.S. PTO
09/26/00

Inventorship..... Vetrivelkumaran et al.
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-658US
Title: Systems and Methods for Controlling the Number of Clients that Access a Server

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks,
Washington, D.C. 20231

From: James R. Banowsky (Tel. 509-324-9256; Fax 509-323-8979)
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Specification—title page, plus 39 pages, including 51 claims and Abstract
2. Transmittal letter including Certificate of Express Mailing
3. 4 Sheets Formal Drawings (Figs. 1-4)
4. Return Post Card

Large Entity Status [x] Small Entity Status []

Date: 9/26/00

By: James R. Banowsky
James R. Banowsky
Reg. No. 37,773

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL685271515

Date: 9/26/2000

By: Dana L. Calhoun
Dana L. Calhoun

EL685271515

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Systems and Methods For Controlling The
Number Of Clients That Access A Server**

Inventors:

Vellore T. Vettrivelkumaran
Raju Gulabani
Steve Falcon
Neel Malik

ATTORNEY'S DOCKET NO. MS1-658US

1 **TECHNICAL FIELD**

2 The systems and methods described herein relate to server-client network
3 systems. More particularly, the described implementations relate to controlling
4 the number of unique clients that can access server software in a server-client
5 network.

6 **BACKGROUND**

7 Developers of operating systems and software applications require that
8 users of these systems have a license to use the system. The license indicates that
9 the user has paid for the right to use the software. Business enterprises that license
10 software for use in enterprise systems usually enter into a concurrent access
11 license with the software developer. A concurrent users license allows a limited
12 number of unique users to use the software. Such licenses, however, can be
13 virtually impossible to enforce.

14 Some network software systems are configured to allow only a certain
15 number of users to access the software at the same time. For example, an
16 operating system may be configured to allow only ten users to access the operating
17 system at any given time. However, limiting access to ten users at the same time
18 does not necessarily limit access to the operating system to ten unique users.
19 Typical use of a system will see many users logging on and off a system for
20 various periods throughout a day. One user may log onto the system for ten
21 minutes and log off. Another user may log on for an hour before logging off.
22 Therefore, it is possible that many more than ten users may access a system even
23 though no more than ten users are logged onto the system at any given time.

1 Furthermore, software developers sometimes develop smaller versions of
2 large software packages. These smaller versions - which are less expensive than
3 the larger versions - are targeted for small businesses that do not need the
4 capabilities of the larger version or that may not be able to afford the larger
5 systems. If the software developer cannot limit the number of unique users that
6 access the system, then a larger enterprise may use a version that was designed for
7 a smaller enterprise and avoid using a more costly software package. Such misuse
8 by an enterprise harms the software developer.

9

10 **SUMMARY**

11 Methods and systems are described herein that limit the number of clients
12 that can access server software to a pre-defined limit. A server operating system -
13 or other software system running on a server - includes a communications protocol
14 filter that monitors transmissions between the server and multiple clients
15 connected to the server via a network.

16 When the communications filter detects a packet header that identifies a
17 client attempting to connect to the server, the communications filter identifies a
18 network address associated with the client from the packet and attempts to locate
19 the network address in a table that contains a network address for each client that
20 has previously accessed the server. If the network address for the client is in the
21 table, then the server processes the communications packets received from the
22 client.

23 If the network address for the client is not listed in the table, then the server
24 references a client limit field in server memory. The client limit field contains a
25 configurable client limit value that denotes the number of clients that are allowed

1 to access the server. The client limit value is encrypted to prevent illegal alteration
2 of the client limit. If the number of entries in the table is less than the value in the
3 client limit field (i.e., the number of clients having access to the server is less than
4 the allowable limit), then the client is allowed to access the server and the network
5 address for the client is added to the table.

6 In some systems, a network address that is assigned to a client may expire
7 after a certain period, such as when the client has been inactive for a certain period
8 of time. When the client becomes active again, a new network address is assigned
9 to the client. Often, the new network address will be identical to the original
10 network address, but there are cases when the network addresses are different.

11 To accommodate these types of systems, access to the server is not simply
12 denied if the number of client network addresses in the table meets or exceeds the
13 client limit. If the number of network addresses in the table is greater than or
14 equal to the client limit when a client attempts to access the server, and the client's
15 network address is not in the table, the communications filter determines if the
16 client requesting server access is a previous client that has a new network address.
17 In one implementation, the communications filter transmits a signal to each
18 network address that is listed in the table. If a client at a network address does not
19 respond to the signal, the communications filter assumes that the network address
20 that does not provide a response is an old network address of the client attempting
21 to access the system. Therefore, it removes the old network address from the
22 table, inserts the network address of the client requesting access into the table, and
23 allows the client to access the server. If, however, all network addresses respond
24 to the signal, then the client limit has been reached and allowing access to the
25 requesting client would exceed the limit. The client is, therefore, denied access to

1 the server and a ‘limit exceeded’ event occurs. Upon occurrence of this event, a
2 signal is sent to indicate that the client limit has been exceeded and the packet is
3 not processed.

4

5 **BRIEF DESCRIPTION OF THE DRAWINGS**

6 A more complete understanding of exemplary methods and arrangements
7 of the present invention may be had by reference to the following detailed
8 description when taken in conjunction with the accompanying drawings wherein:

9 Fig. 1 is an exemplary computer system on which the present invention
10 may be implemented.

11 Fig. 2 is an illustration of a server having a communications filter, the
12 server communicating with several clients over a network.

13 Fig. 3 is a block diagram of a server-client network system implemented in
14 accordance with the described embodiments.

15 Fig. 4 is a flow diagram outlining a method for controlling the number of
16 clients having access to a server in a server-client network system.

1 **DETAILED DESCRIPTION**

2 The invention is illustrated in the drawings as being implemented in a
3 suitable computing environment. Although not required, the invention will be
4 described in the general context of computer-executable instructions, such as
5 program modules, to be executed by a computing device, such as a personal
6 computer or a hand-held computer or electronic device. Generally, program
7 modules include routines, programs, objects, components, data structures, etc. that
8 perform particular tasks or implement particular abstract data types. Moreover,
9 those skilled in the art will appreciate that the invention may be practiced with
10 other computer system configurations, including multi-processor systems,
11 microprocessor-based or programmable consumer electronics, network PCs,
12 minicomputers, mainframe computers, and the like. The invention may also be
13 practiced in distributed computing environments where tasks are performed by
14 remote processing devices that are linked through a communications network. In
15 a distributed computing environment, program modules may be located in both
16 local and remote memory storage devices.

17

18 **Exemplary Computer Environment**

19 The various components and functionality described herein are
20 implemented with a number of individual computers. Fig. 1 shows components of
21 typical example of such a computer, referred by to reference numeral 100. The
22 components shown in Fig. 1 are only examples, and are not intended to suggest
23 any limitation as to the scope of the functionality of the invention; the invention is
24 not necessarily dependent on the features shown in Fig. 1.

Generally, various different general purpose or special purpose computing system configurations can be used. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The functionality of the computers is embodied in many cases by computer-executable instructions, such as program modules, that are executed by the computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Tasks might also be performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media.

The instructions and/or program modules are stored at different times in the various computer-readable media that are either part of the computer or that can be read by the computer. Programs are typically distributed, for example, on floppy disks, CD-ROMs, DVD, or some form of communication media such as a modulated signal. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable media when such media contain instructions programs, and/or modules for implementing the steps

1 described below in conjunction with a microprocessor or other data processors.
2 The invention also includes the computer itself when programmed according to
3 the methods and techniques described below.

4 For purposes of illustration, programs and other executable program
5 components such as the operating system are illustrated herein as discrete blocks,
6 although it is recognized that such programs and components reside at various
7 times in different storage components of the computer, and are executed by the
8 data processor(s) of the computer.

9 With reference to Fig. 1, the components of computer 100 may include, but
10 are not limited to, a processing unit 120, a system memory 130, and a system bus
11 121 that couples various system components including the system memory to the
12 processing unit 120. The system bus 121 may be any of several types of bus
13 structures including a memory bus or memory controller, a peripheral bus, and a
14 local bus using any of a variety of bus architectures. By way of example, and not
15 limitation, such architectures include Industry Standard Architecture (ISA) bus,
16 Micro Channel Architecture (MCA) bus, Enhanced ISA (EISAA) bus, Video
17 Electronics Standards Association (VESA) local bus, and Peripheral Component
18 Interconnect (PCI) bus also known as the Mezzanine bus.

19 Computer 100 typically includes a variety of computer-readable media.
20 Computer-readable media can be any available media that can be accessed by
21 computer 100 and includes both volatile and nonvolatile media, removable and
22 non-removable media. By way of example, and not limitation, computer-readable
23 media may comprise computer storage media and communication media.
24 “Computer storage media” includes both volatile and nonvolatile, removable and
25 non-removable media implemented in any method or technology for storage of

information such as computer-readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more if its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 100, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

1 The computer 100 may also include other removable/non-removable,
2 volatile/nonvolatile computer storage media. By way of example only, Fig. 1
3 illustrates a hard disk drive 141 that reads from or writes to non-removable,
4 nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to
5 a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that
6 reads from or writes to a removable, nonvolatile optical disk 156 such as a CD
7 ROM or other optical media. Other removable/non-removable,
8 volatile/nonvolatile computer storage media that can be used in the exemplary
9 operating environment include, but are not limited to, magnetic tape cassettes,
10 flash memory cards, digital versatile disks, digital video tape, solid state RAM,
11 solid state ROM, and the like. The hard disk drive 141 is typically connected to
12 the system bus 121 through a non-removable memory interface such as interface
13 140, and magnetic disk drive 151 and optical disk drive 155 are typically
14 connected to the system bus 121 by a removable memory interface such as
15 interface 150.

16 The drives and their associated computer storage media discussed above
17 and illustrated in Fig. 1 provide storage of computer-readable instructions, data
18 structures, program modules, and other data for computer 100. In Fig. 1, for
19 example, hard disk drive 141 is illustrated as storing operating system 144,
20 application programs 145, other program modules 146, and program data 147.
21 Note that these components can either be the same as or different from operating
22 system 134, application programs 135, other program modules 136, and program
23 data 137. Operating system 144, application programs 145, other program
24 modules 146, and program data 147 are given different numbers here to illustrate
25 that, at a minimum, they are different copies. A user may enter commands and

1 information into the computer 100 through input devices such as a keyboard 162
2 and pointing device 161, commonly referred to as a mouse, trackball, or touch
3 pad. Other input devices (not shown) may include a microphone, joystick, game
4 pad, satellite dish, scanner, or the like. These and other input devices are often
5 connected to the processing unit 120 through a user input interface 160 that is
6 coupled to the system bus, but may be connected by other interface and bus
7 structures, such as a parallel port, game port, or a universal serial bus (USB). A
8 monitor 191 or other type of display device is also connected to the system bus
9 121 via an interface, such as a video interface 190. In addition to the monitor,
10 computers may also include other peripheral output devices such as speakers 197
11 and printer 196, which may be connected through an output peripheral interface
12 195.

13 The computer may operate in a networked environment using logical
14 connections to one or more remote computers, such as a remote computer 180.
15 The remote computer 180 may be a personal computer, a server, a router, a
16 network PC, a peer device or other common network node, and typically includes
17 many or all of the elements described above relative to computer 100, although
18 only a memory storage device 181 has been illustrated in Fig. 1. The logical
19 connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide
20 area network (WAN) 173, but may also include other networks. Such networking
21 environments are commonplace in offices, enterprise-wide computer networks,
22 intranets, and the Internet.

23 When used in a LAN networking environment, the computer 100 is
24 connected to the LAN 171 through a network interface or adapter 170. When used
25 in a WAN networking environment, the computer 100 typically includes a modem

1 172 or other means for establishing communications over the WAN 173, such as
2 the Internet. The modem 172, which may be internal or external, may be
3 connected to the system bus 121 via the user input interface 160, or other
4 appropriate mechanism. In a networked environment, program modules depicted
5 relative to the computer 100, or portions thereof, may be stored in the remote
6 memory storage device. By way of example, and not limitation, Fig. 1 illustrates
7 remote application programs 185 as residing on memory device 181. It will be
8 appreciated that the network connections shown are exemplary and other means of
9 establishing a communications link between the computers may be used.

10 Fig. 2 is a simplified illustration of a computer network system 200 that
11 implements one or more of the described implementations. The computer network
12 system 200 includes a server 202 and several client computers that communicate
13 with the server 202 over a network 204. For purposes of the present discussion,
14 the network 204 is a local area network (LAN), although other types of networks
15 may be used. Client A 206, Client B 208, Client C 210 and Client D 212 are
16 connected to the network 204. Communications with the server 202 are directed
17 through a communications filter 214, which controls the number of clients that can
18 access the server or utilize software on the server. The communications filter 214
19 monitors communications between the server 202 and the client computers 206 –
20 212 and is configured to identify a network address for each of the clients 206 –
21 212 that communicate with the server 202.

22 Although the communications filter 214 is shown as being separate from
23 the server 202, the communications filter 214 may be located on the server 202 or
24 on a peripheral unit connected to the server 202. Generally, the communications
25 filter 214 may be located anywhere that it can monitor communications between

1 the server 202 and the clients 206 – 212 and where a client cannot bypass the
2 communications filter 214 to communicate with the server 202.

3 The communications filter 214 may control access to the server 202 itself
4 (i.e., to a server operating system), to a software application stored on the server
5 202, or to a hardware device (not shown) that is connected to the server 202. The
6 present discussion will focus on a communications filter that is configured to limit
7 the number of unique clients that can access and utilize a server operating system.
8 Also, for purposes of the present discussion, reference made to accessing the
9 server will mean accessing an operating system on the server 202.

10 Fig. 3 is a block diagram of a server-client network system 300
11 implemented in accordance with the described embodiments. The system 300
12 includes a server 302 that communicates with a network 306. Client A 308, client
13 B 310, client C 312 and client D 314 communicate with the server 302 via the
14 network 304. It is noted that, although only four clients are shown that
15 communicate with the server 302, virtually any number of clients can
16 communicate with the server 302. The number of clients accessing the server 302
17 is limited in the present invention only by a communications filter, which will be
18 discussed in greater detail below.

19 Client A 308 includes a network interface card 318 (NIC) that is used to
20 connect client A 308 to the network 306. Client A 308 also has a network address
21 320 that uniquely identifies client A 308 on the network 306. Client B 310
22 includes a network interface card 322 that connects client B 310 to the network
23 306. Client B 310 also has a network address 324 that uniquely identifies client B
24 310 on the network 306. Client C 312 includes a network interface card 326 to
25 connect client C 312 to the network 306. Client C 312 also has a network address

1 328 that uniquely identifies client C 312 on the network 306. Client D 314
2 includes a network interface card 330 and has a network address 332 that uniquely
3 identifies client D 314 on the network 306.

4 The server 302 includes a processor 334, a network interface card 336 and
5 memory 338. The processor 334 is a central processing unit that is configured to
6 execute processor-executable instructions. The network interface card 336
7 provides an interface between the server 302 and the network 306. The server 302
8 communicates with the network 306 using standard TCP/IP (Transmission Control
9 Protocol/Internet Protocol) communications protocol.

10 It is noted that, although the server 302 is shown communicating via the
11 network 306 using TCP/IP, the server 302 may also communicate over any
12 network - such as a wide area network (WAN) - utilizing any appropriate protocol.
13 The server 302 may also be configured to use a method other than a network
14 interface card to interface with the network 306. For example, the server 302 may
15 utilize a modem (not shown) to communicate with client computers in a wide area
16 network.

17 The memory 338 includes an operating system 340, which has a global
18 system registry 342. A client limit 344 is stored in the registry 342 and is,
19 therefore, available for retrieval by various software applications. The client limit
20 344, as will be discussed in greater detail below, is a value that specifies a
21 maximum number of unique clients that can access the server 302. For purposes
22 of discussion, accessing the server 302 means accessing and utilizing the operating
23 system 340, although the described implementations could be used to limit access
24 to the server 302, a software application resident on the server 302, or some piece
25 of electronic hardware unit. The client limit 344 is the number of users allowed by,

1 for example, a concurrent users license. The client limit 344 is preferably
2 configurable, so that a software designer or vendor can customize the client limit
3 for each particular server. The configurable client limit 344 can be set to different
4 limits on different server products.

5 The memory 338 also includes a protocol stack 346 that is used to process
6 communication protocol commands received from one or more clients 308 – 314.
7 The protocol stack 346 includes related communication protocol layers stacked on
8 top of each other (data products, sessions, etc.)

9 A decryption module 348 resides in the memory 338 and is used when the
10 client limit 344 is stored in an encrypted form. The encrypted client limit 344 is
11 decrypted by the decryption module 348 and is stored in the memory as client
12 limit 350. The decrypted client limit 350 is used in subsequent processing. In an
13 alternative implementation, the decryption module 348 decrypts the encrypted
14 client limit 344 whenever the client limit 344 is used in processing. However, it is
15 preferable to utilize a decrypted client limit 350 in processing, as this economizes
16 computational overhead.

17 The memory 338 also includes a communications filter 352 that is
18 configured to limit the number of clients that can access the server 302 to the
19 value indicated by the client limit 350. The communications filter 352 consists of
20 a communications filter driver 354 and a communications filter controller 356.
21 The communications filter driver 354, as will be discussed below, is configured to
22 signal an event when the client limit 350 is exceeded. The communications filter
23 controller 356 initializes the communications filter driver 354 and retrieves the
24 client limit 350 (or the client limit 344 in the registry 342). The communications
25 filter controller 356 is additionally configured to send a signal to all clients that

1 have previously accessed the operating system 340 and monitor for a response
2 from each client. The purpose of this function will be discussed in detail, below.

3 It is noted that the preferred implementation, as described below, will
4 identify the tasks that are allocated to and performed by the communications filter
5 controller 356. However, it is noted that the communications filter controller 356
6 may also be configured to perform other tasks described in the following
7 discussion, although such other tasks may not be specifically recited as being
8 performed by the communications filter controller 356.

9 The communications filter driver 354, when initialized with the client limit
10 350, creates a client table 358 having a number of entries equal to the number of
11 unique users allowed to access the operating system 340 under the applicable users
12 license. The communications filter driver 354 then registers with the
13 communications stack 346 to receive notification on every incoming packet on the
14 local network interface card 336 (the local NIC is the network interface card
15 through which clients on the network 306 connect to the server 302). The
16 remaining functions of the communications filter driver 354 will be discussed in
17 greater detail, below, with reference to Fig. 4.

18 The client table 358 has one entry available for each unique user that
19 accesses the operating system 340. For example, if the client limit 350 is twenty-
20 five (25), then the client table 358 will consist of twenty-five (25) entries, or rows.
21 As shown in Fig. 3, the client table 358 comprises three (3) rows: row 360, row
22 362 and row 364. For purposes of this example, it is assumed that the client limit
23 is three (3), although the client limit 350 could be virtually any number. In the
24 present example, row 360 contains a network address associated with and uniquely
25 identifying client A 308. The network address associated with client A 308 is

1 64.236.8.10. Row 362 contains a network address associated with client B 310,
2 and row 364 contains a network address associated with client C 312. The
3 network address for client B 310 is 64.220.16.27, and the network address
4 assigned to client C 312 is 32.99.128.16.

5 It is noted that the network addresses stored in row 360, row 362 and row
6 364 will not actually be stored in the rows until client A, client B and client C are
7 granted access to the operating system 340. However, for convenience purposes,
8 the network addresses are shown stored in the rows 360 – 364. The process by
9 which the network addresses are stored is discussed in detail, below, with
10 reference to Fig. 4.

11 Fig. 4 is a flow diagram that depicts a method in accordance with the
12 present invention. At step 400, the communications filter driver 354 monitors
13 incoming communications packets that are sent across the local network interface
14 card 336. If a communications packet is a session initiation packet - such as a TCP SYN (TCP synchronization) packet - (“Yes” branch, step 402), then
15 the network address is identified from the communications packet at step 404. If
16 the communications packet is not a session initiation packet - such as a TCP SYN
17 packet - (“No” branch, step 404), then it is assumed that the client has previously
18 accessed the server 302, and the packet is processed at step 416.
19

20 It is noted that the communications filter driver 354 could examine all
21 communications packets received from a client. However, for performance
22 reasons, in the preferred implementation only the session initiation packets are
23 inspected to determine the network address of a sending client.

24 Once, the network address has been determined from the packet, the
25 communications filter driver 354 determines if the network address is listed in the

1 client table 358. For discussion purposes, assume that at this point, the client table
2 358 contains no entries and that client A 308 has sent a TCP SYN packet to the
3 server 302. At step 406, the communications filter driver 354 searches the client
4 table 358 for the network address 320 of client A 308 (64.236.8.100). If the
5 network address for client A 308 is found in the client table 358 ("Yes" branch,
6 step 406), then client A 358 is allowed to access the server 302 and the packet is
7 processed at step 416. But in this example, the client table 358 is empty so no
8 match is found for the network address 320 of client A 308 ("No" branch, step
9 406).

10 At step 408, the communications filter driver 354 determines if the
11 maximum number of clients that are allowed to access the operating system 340
12 has been reached. The client limit 350 is retrieved from memory 338. If the
13 number of entries in the client table 358 is less than the client limit 350 ("No"
14 branch, step 408), then the network address 320 for client A 308 is stored in the
15 client table 358 at row 360 (step 410) and the packet is processed at step 416.

16 At this point, client A 308 has access to the server operating system 340.
17 Continuing with the example, now assume that client B 310 attempts to access the
18 operating system 344 of the server 302. At step 402, the communications filter
19 driver 354 detects a data packet that includes the network address 324 of client B
20 310. The network address 324 is identified at step 404 and, at step 406, the
21 communications filter driver 354 compares the network address 324 for client B
22 310 (64.220.16.27) to the entries in the client table 358 in an attempt to find a
23 match. Since client B 310 has not previously accessed the operating system 344,
24 the communications filter determines if the client limit 350 has been reached by
25 comparing the number of entries in the client table 358 to the client limit 350. In

1 this example, the client limit 350 is three (3) and the number of entries in the client
2 table 358 is one (client A). Therefore, the limit has not been reached (“No”
3 branch, step 408) and the network address 324 for client B 310 is stored in the
4 client table 358 at step 410. The communications packets received from client B
5 310 are processed at step 416.

6 At this point, client A 308 and client B 310 have access to the server
7 operating system 340, and the network addresses 320, 324 for client A 308 and
8 client B 310 are stored in the client table 358. Continuing with the example, client
9 C 312 attempts to access the operating system 344 of the server 302. At step 402,
10 the communications filter driver 354 detects a data packet that includes the
11 network address 328 of client C 312. The network address 328 is identified at step
12 404 and, at step 406, the communications filter driver 354 compares the network
13 address 328 for client C 312 (126359.44.33) with the entries in the client table 358
14 to find a match. Since client C 312 has not previously accessed the operating
15 system 344, the communications filter determines if the client limit 350 has been
16 reached by comparing the number of entries in the client table 358 to the client
17 limit 350. The client limit 350 is three (3) and the number of entries in the client
18 table 358 is now two (clients A and B). Therefore, the client limit has not been
19 reached (“No” branch, step 408) and the network address 328 for client C 312 is
20 stored in the client table 358 at step 410. The communications packets received
21 from client C 312 are processed at step 416.

22 Assume now that client D 314 attempts to communicate with the operating
23 system 344 of the server 302. At step 402, the communications filter driver 354
24 detects a TCP SYN packet sent from client D 314 and the network address 332 for
25 client D 314 is identified. The network address 332 for client D 314 is not in the

1 client table 358 (“No” branch, step 406), so the client limit is checked at step 408.
2 Now, the number of entries (or network addresses) in the client table 358 is three,
3 which is equal to the client limit 350. Therefore, client D 314 is not immediately
4 allowed access to the operating system 344.

5 As previously discussed, there are some systems that are configured to
6 reassign a network address for a client when the network address for the client has
7 expired for one of various reasons. For example, a system might reclaim a
8 network address from a client if the client has not been active for a certain period
9 of time. If the client becomes active again at a later time, the system assigns a
10 new network address to the client. This new network address may be identical to
11 the original network address, but it can be a different network address. Therefore,
12 the present invention must accommodate this type of scheme.

13 At step 412, the communications filter driver 354 attempts to determine if
14 client D 316 has previously accessed the operating system 344 using a different
15 network address. If client D 316 has previously accessed the server 302 using a
16 different network address (“Yes” branch, step 412), then the new network address
17 used by client D 314 is substituted for the old network address used by client D
18 314 (step 414). But if client D 314 has not previously accessed the server 302,
19 then at step 418, a limit exceeded event is initiated wherein the communications
20 filter driver 354 notifies the communications filter controller 356 that the limit has
21 been exceeded. The communications filter controller 356 sends a “limit
22 exceeded” signal to the operating system 340 (and which is ultimately sent to the
23 client) and the packet is dropped. In other words, client D 316 is not allowed to
24 communicate with the server 302.

25

1 One way in which the communications filter 352 determines if the new
2 client (client D 316) has previously accessed the server 302 is by the
3 communications filter controller 356 sending a signal to each network address
4 listed in the client table 358. If a client at one network address fails to
5 acknowledge the signal, then the communications filter controller 356 assumes
6 that the network address is no longer valid and, therefore, another client may
7 access the server 302 within the client limit 350. When a response is not received
8 from a network address, the communications filter controller 356 removes the
9 non-responsive network address from the client table 358 and inserts the new
10 network address in its place (step 414). Therefore, the client limit 350 is still not
11 exceeded.

12 It is noted that a client may not respond because it has been temporarily
13 shut down. If so, then it is theoretically possible to exceed the client limit.
14 However, it would be very inconvenient for users to do this simply to exceed a
15 concurrent users license and it is only a small probability that such a technique
16 would, in practice, actually be used to exceed the limit. That notwithstanding, the
17 described implementations still provide a reliable, practical technique for keeping
18 the number of clients that access the server within the client limit.

19

20 Conclusion

21 The systems and methods described herein provide a practical way to limit
22 the number of clients that can access a server in a server-client network system.
23 The server keeps track of each network address from which it is accessed and
24 denies access to any client that attempts to access the server after the maximum
25 number of clients have accessed the server.

1 Systems that allow network address to expire and that reassign network
2 addresses to existing clients can be implemented as well. When a client attempts
3 to access the server when the client limit has been reached, the server signals each
4 network address that identifies a client having access to the server. If a network
5 address does not acknowledge the signal, then the server assumes that the network
6 address is no longer in use, and the server allows another client to access the
7 server.

8 Although details of specific implementations and embodiments are
9 described above, such details are intended to satisfy statutory disclosure
10 obligations rather than to limit the scope of the following claims. Thus, the
11 invention as defined by the claims is not limited to the specific features described
12 above. Rather, the invention is claimed in any of its forms or modifications that
13 fall within the proper scope of the appended claims, appropriately interpreted in
14 accordance with the doctrine of equivalents.

DRAFT - DO NOT CITE

1 **CLAIMS**

2

3 **1.** An Internet protocol (IP) filter, comprising processor-executable
4 instructions that, when executed on a processor, perform the following steps:

5 monitoring Internet protocol data packets transmitted from one or more
6 clients to a server;

7 obtaining a network address from an IP data packet transmitted by a client;
8 and

9 processing IP data packets from the client if a Network address that is
10 uniquely associated with the client is stored in a client table.

11

12 **2.** The Internet protocol filter as recited in claim 1, further comprising
13 processor-executable instructions that, when executed on a processor, perform the
14 following steps:

15 if the Network address is not stored in the client table, retrieving a client
16 limit value from a client limit field, the client limit value indicating a maximum
17 number of unique clients for which IP data packets can be processed;

18 processing IP data packets from the client if the number of Network
19 addresses in the client table is less than the client limit value; and

20 storing the Network address in the client table.

1 **3.** The Internet protocol filter as recited in claim 1, wherein the client is
2 a first client and the Network address is a first Network address, the Internet
3 protocol filter further comprising processor-executable instructions that, when
4 executed on a processor, perform the following steps:

5 if the first Network address is not stored in the client table, retrieving a
6 client limit value from a client limit field, the client limit value indicating a
7 maximum number of unique clients for which IP data packets can be processed;

8 if the number of Network addresses in the client table is greater than or
9 equal to the client limit value, determining if the first client is represented in the
10 client table by a second Network address that is different from the first Network
11 address; and

12 processing IP data packets from the first client if the second Network
13 address is found in the client table.

14
15 **4.** The Internet protocol filter as recited in claim 3, further comprising
16 processor-executable instructions that, when executed on a processor, perform the
17 following steps:

18 removing the second Network address from the client table; and

19 inserting the first Network address into the client table.

1 **5.** The Internet protocol filter as recited in claim 1, further comprising
2 processor-executable instructions that, when executed on a processor, perform the
3 following steps:

4 if the first Network address is not stored in the client table, retrieving a
5 client limit value from a client limit field, the client limit value indicating a
6 maximum number of unique clients for which IP data packets can be processed;

7 if the number of Network addresses in the client table is greater than or
8 equal to the client limit value, transmitting a signal to each Network address listed
9 in the client table; and

10 if a client at a second Network addresses does not respond to the signal,
11 removing the second Network address from the client table, inserting the first
12 Network address into the client table and processing IP data packets from the first
13 client.

14

15 **6.** The Internet protocol filter as recited in claim 5, further comprising
16 processor-executable instructions that, when executed on a processor, perform the
17 following steps:

18 removing the second Network address from the client table; and

19 inserting the first Network address into the client table.

1 7. A method, comprising:

2 detecting when a current client attempts to establish a connection with a
3 server;

4 determining a unique client identifier that is associated with the current
5 client;

6 determining if a total number of previous clients having access to the server
7 is less than a client limit;

8 determining if the current client has previously been allowed to access the
9 server;

10 providing access to the server if the total number of previous clients having
11 access to the server is less than a client limit;

12 providing access to the server if the total number of previous clients is
13 greater than or equal to the client limit and if the current client has previously been
14 allowed to access the server; and

15 storing the unique client identifier associated with the current client in
16 memory if access is provided to the current client.

17

18 8. The method as recited in claim 7, wherein the determining if a total
19 number of previous clients having access to the server is less than a client limit
20 further comprises:

21 determining how many unique identifiers are stored in memory; and

22 comparing the number of unique identifiers in memory with the client limit.

1 9. The method as recited in claim 7, wherein the determining if the
2 current client has previously been allowed to access the server is only performed if
3 the total number of previous clients having access to the server is greater than or
4 equal to the client limit.

5
6 10. The method as recited in claim 7, wherein the determining if the
7 current client has previously been allowed to access the server further comprises:

8 comparing the unique identifier of the current client with the unique
9 identifiers of each previous client that has been allowed to access the server;

10 determining that the current client has previously been allowed to access
11 the server if the current client identifier matches a previous client identifier.

12
13 11. The method as recited in claim 7, wherein the determining if the
14 current client has previously been allowed to access the server further comprises:

15 transmitting a signal to each previous client that has been allowed to access
16 the server; and

17 determining that the current client has previously been allowed to access
18 the server if at least one of the previous clients fails to acknowledge the signal.

19
20 12. The method as recited in claim 7, further comprising:

21 pre-configuring the client limit; and

22 storing the client limit in memory.

1 **13.** The method as recited in claim 12, wherein the client limit has a pre-
2 defined maximum to which it may be configured.

3
4 **14.** The method as recited in claim 7, further comprising:
5 pre-configuring the client limit;
6 encrypting the client limit; and
7 storing the encrypted client limit in memory.

8
9 **15.** The method as recited in claim 7, further comprising:
10 retrieving an encrypted client limit; and
11 decrypting the encrypted client limit to derive the client limit.

12
13 **16.** The method as recited in claim 7, wherein the determining the
14 unique client identifier that is associated with the current client further comprises
15 identifying an Internet protocol address from a data packet transmitted by the
16 current client.

17
18 **17.** The method as recited in claim 7, further comprising storing the
19 unique client identifiers in a client table in memory.

20
21 **18.** The method as recited in claim 7, wherein the client identifier is a
22 network address.

1 **19.** A server that provides access to a limited number of clients,
2 comprising:

3 memory;

4 a network interface configured to handle communications between the
5 server and a plurality of clients;

6 an operating system stored in the memory;

7 a client limit stored in the memory, the client limit denoting a number of
8 unique clients that are allowed to access the server;

9 an IP stack in the memory that is used to process data packets transmitted
10 from clients;

11 a client table in the memory for storing a unique Network address for each
12 client that accesses the server; and

13 a communications filter configured to allow access to a first client if the
14 total number of clients that have accessed the server is less than the client limit, or
15 if the total number of clients that have accessed the server is greater than or equal
16 to the client limit and the first client has previously accessed the server.

17
18 **20.** The server as recited in claim 19, wherein the Communications filter
19 is further configured to search the client table for a first Network address
20 associated with the first client and determine that the first client has previously
21 accessed the server if the first Network address is found in the client table.

1 **21.** The server as recited in claim 19, wherein the Communications filter
2 is further configured to search the client table for a second Network address
3 associated with the first client and determine that the first client has previously
4 accessed the server if the second Network address is found in the client table.

5

6 **22.** The server as recited in claim 21, wherein the Communications filter
7 is further configured to determine the second Network address by signaling each
8 Network address listed in the client table and determine that the second Network
9 address is a network address listed in the table that does not acknowledge the
10 signal.

11

12 **23.** The server as recited in claim 19, wherein the client limit is
13 configurable.

14

15 **24.** The server as recited in claim 19, wherein the Communications filter
16 is further configured to signal that the client limit has been exceeded and to deny
17 server access to the first client if the total number of clients that have accessed the
18 server is greater than or equal to the client limit, and the first client has not
19 previously accessed the server.

20

21 **25.** The server as recited in claim 19, wherein the client limit is
22 encrypted, the server further comprising a decryption module configured to
23 decrypt the encrypted client limit.

1 **26.** A method for providing server access to a limited number of clients,
2 the method comprising:
3 monitoring TCP/IP packets sent from a plurality of clients to a server;
4 obtaining a unique Network address for each client from one or more
5 packets transmitted by the client;
6 storing the Network address of each client that accesses the server;
7 determining if a client limit has been reached; and
8 providing access to a first client if the client limit has been reached, or if
9 the first client has previously accessed the server.

10
11 **27.** The method as recited in claim 26, wherein the determining if the
12 client limit has been reached further comprises:
13 determining how many unique clients have accessed the server;
14 comparing the number of unique client with the client limit; and
15 determining that the client limit has been reached if the number of unique
16 clients is greater than or equal to the client limit.

17
18 **28.** The method as recited in claim 26, wherein the determining if the
19 first client has previously accessed the server further comprises:
20 comparing a first Network address that uniquely identifies the first client
21 with a table of stored Network addresses; and
22 determining that the first client has previously accessed the server if the
23 first Network address matches a stored Network address.

1 **29.** The method as recited in claim 26, wherein the determining if the
2 first client has previously accessed the server further comprises:

3 sending a signal to each of multiple Network addresses of clients that have
4 accessed the server; and

5 if there is no response to one of the signals, determining that the first client
6 has previously accessed the server using the Network address of the client from
7 which there was no response detected.

8
9 **30.** The method as recited in claim 26, further comprising:

10 retrieving an encrypted client limit; and

11 decrypting the encrypted client limit to derive the client limit.

12
13 **31.** An operating system stored on a computer-readable medium, the
14 operating system comprising:

15 an IP stack for processing Internet protocol data packets received from
16 multiple clients;

17 a client limit field containing a client limit value that denotes a maximum
18 number of clients that may access the IP stack;

19 a client table containing a unique Network address for each client that has
20 accessed the operating system; and

21 a communications filter configured to determine a first Network address of
22 a first client attempting to access the operating system, search the client table for
23 the first Network address, and allow the first client to access the operating system
24 if the first Network address is found in the client table.

1 **32.** The operating system as recited in claim 31, wherein the
2 Communications filter is further configured to:

3 allow the first client to access the operating system if the number of
4 Network addresses in the client table is less than the client limit value; and
5 store the first Network address in the client table if the first client is allowed
6 to access the operating system.

7

8 **33.** The operating system as recited in claim 31, wherein the
9 Communications filter is further configured to allow the first client to access the
10 operating system if the number of Network addresses in the client table is greater
11 than or equal to the client limit value and the first client has previously accessed
12 the operating system using a second Network address that is stored in the client
13 table.

14

15 **34.** The operating system as recited in claim 33, wherein the
16 Communications filter is further configured to:

17 transmit a signal to each Network address listed in the client table;
18 monitor for an acknowledgement to each signal; and
19 if an acknowledgement is not received from a network address in the client
20 table, determining that the non-acknowledging Network address is the second
21 Network address used by the first client.

1 **35.** The operating system as recited in claim 34, wherein the
2 Communications filter is further configured to replace the second Network address
3 in the client table with the first Network address.

4

5 **36.** The operating system as recited in claim 31, wherein the client limit
6 value is encrypted, and the operating system further comprises a decryption
7 module that is configured to decrypt the client limit value.

8

9 **37.** A computer-readable medium comprising computer-executable
10 instructions that, when executed on a computer, perform the following steps:

11 determining a first Internet Protocol (IP) address transmitted from a first
12 client to a server;

13 searching a client table for the first Network address; and

14 allowing the first client to access the server if the first Network address is
15 found in the client table.

16

17 **38.** The computer-readable medium as recited in claim 37, further
18 comprising computer-executable instructions that, when executed on a computer,
19 perform the following steps:

20 determining if a client limit has been reached, the client limit indicating a
21 total number of clients that can access the server;

22 allowing the first client to access the server if the client limit has not been
23 reached; and

24 inserting the first Network address into the client table.

1 **39.** The computer-readable medium as recited in claim 37, further
2 comprising computer-executable instructions that, when executed on a computer,
3 perform the following steps:

4 transmitting a signal to each Network address listed in the client table; and
5 if there is no response from one of the Network addresses signaled,
6 allowing the first client to access the server, removing the non-responsive Network
7 address from the client table, and inserting the first Network address into the client
8 table.

9
10 **40.** A computer system, comprising:

11 a processor;
12 a network interface card to handle communications with multiple clients;
13 memory;
14 a global system registry;
15 a client table having one entry for each client allowed to access the system,
16 each entry including a unique Internet protocol (IP) address for each client; and
17 a communications filter configured to:

18 retrieve a client limit from the global system registry;
19 determine a first Network address that is associated with a first client
20 attempting to access the system;

21 allow the first client to access the system if the first Network address
22 is stored in the client table or if the number of client table entries is less
23 than the client limit; and

24 store the first Network address in the client table if the first client is
25 allowed to access the system.

1
2 **41.** The computer system as recited in claim 40, wherein the
3 Communications filter is further configured to allow the first client to access the
4 system if the number of entries in the client table is greater than or equal to the
5 client limit and if the first client has previously accessed the system.
6

7 **42.** The computer system as recited in claim 41, wherein the
8 Communications filter is further configured to determine if the first client has
9 previously accessed the system if the first Network address is stored in the client
10 table.
11

12 **43.** The computer system as recited in claim 41, wherein the
13 Communications filter is further configured to determine if the first client has
14 previously accessed the system by transmitting a signal to each Network address
15 listed in the client table, monitoring responses to the signals to determine if a
16 client at a second Network address is no longer using the Network address,
17 substituting the first Network address in the table for the second Network address
18 and allowing the first client to access the system if the client at the second
19 Network address does not respond to the signal.
20

21 **44.** The computer system as recited in claim 40, wherein the client limit
22 is encrypted, the computer system further comprising a decryption module
23 configured to decrypt the encrypted client limit.
24
25

1 **45.** A communications protocol filter, comprising processor-executable
2 instructions that, when executed on a processor, perform the following steps:

3 monitoring communications protocol data packets transmitted from one or
4 more clients to a server;

5 obtaining a network address from a communications protocol data packet
6 transmitted by a client; and

7 processing communications protocol data packets from the client if a
8 Network address that is uniquely associated with the client is stored in a client
9 table.

10
11 **46.** The communications protocol filter as recited in claim 45, further
12 comprising processor-executable instructions that, when executed on a processor,
13 perform the following steps:

14 if the Network address is not stored in the client table, retrieving a client
15 limit value from a client limit field, the client limit value indicating a maximum
16 number of unique clients for which communications data packets can be
17 processed;

18 processing communications protocol data packets from the client if the
19 number of Network addresses in the client table is less than the client limit value;
20 and

21 storing the Network address in the client table.

1 **47.** The communications protocol filter as recited in claim 45, wherein
2 the client is a first client and the Network address is a first Network address, the
3 communications protocol filter further comprising processor-executable
4 instructions that, when executed on a processor, perform the following steps:

5 if the first Network address is not stored in the client table, retrieving a
6 client limit value from a client limit field, the client limit value indicating a
7 maximum number of unique clients for which communications protocol data
8 packets can be processed;

9 if the number of Network addresses in the client table is greater than or
10 equal to the client limit value, determining if the first client is represented in the
11 client table by a second Network address that is different from the first Network
12 address; and

13 processing communications protocol data packets from the first client if the
14 second Network address is found in the client table.

15
16 **48.** The communications protocol filter as recited in claim 47, further
17 comprising processor-executable instructions that, when executed on a processor,
18 perform the following steps:

19 removing the second Network address from the client table; and
20 inserting the first Network address into the client table.

1 **49.** The communications protocol filter as recited in claim 45, further
2 comprising processor-executable instructions that, when executed on a processor,
3 perform the following steps:

4 if the first Network address is not stored in the client table, retrieving a
5 client limit value from a client limit field, the client limit value indicating a
6 maximum number of unique clients for which communications protocol data
7 packets can be processed;

8 if the number of Network addresses in the client table is greater than or
9 equal to the client limit value, transmitting a signal to each Network address listed
10 in the client table; and

11 if a client at a second Network addresses does not respond to the signal,
12 removing the second Network address from the client table, inserting the first
13 Network address into the client table and processing communications protocol
14 data packets from the first client.

15
16 **50.** The communications protocol filter as recited in claim 49, further
17 comprising processor-executable instructions that, when executed on a processor,
18 perform the following steps:

19 removing the second Network address from the client table; and

20 inserting the first Network address into the client table.

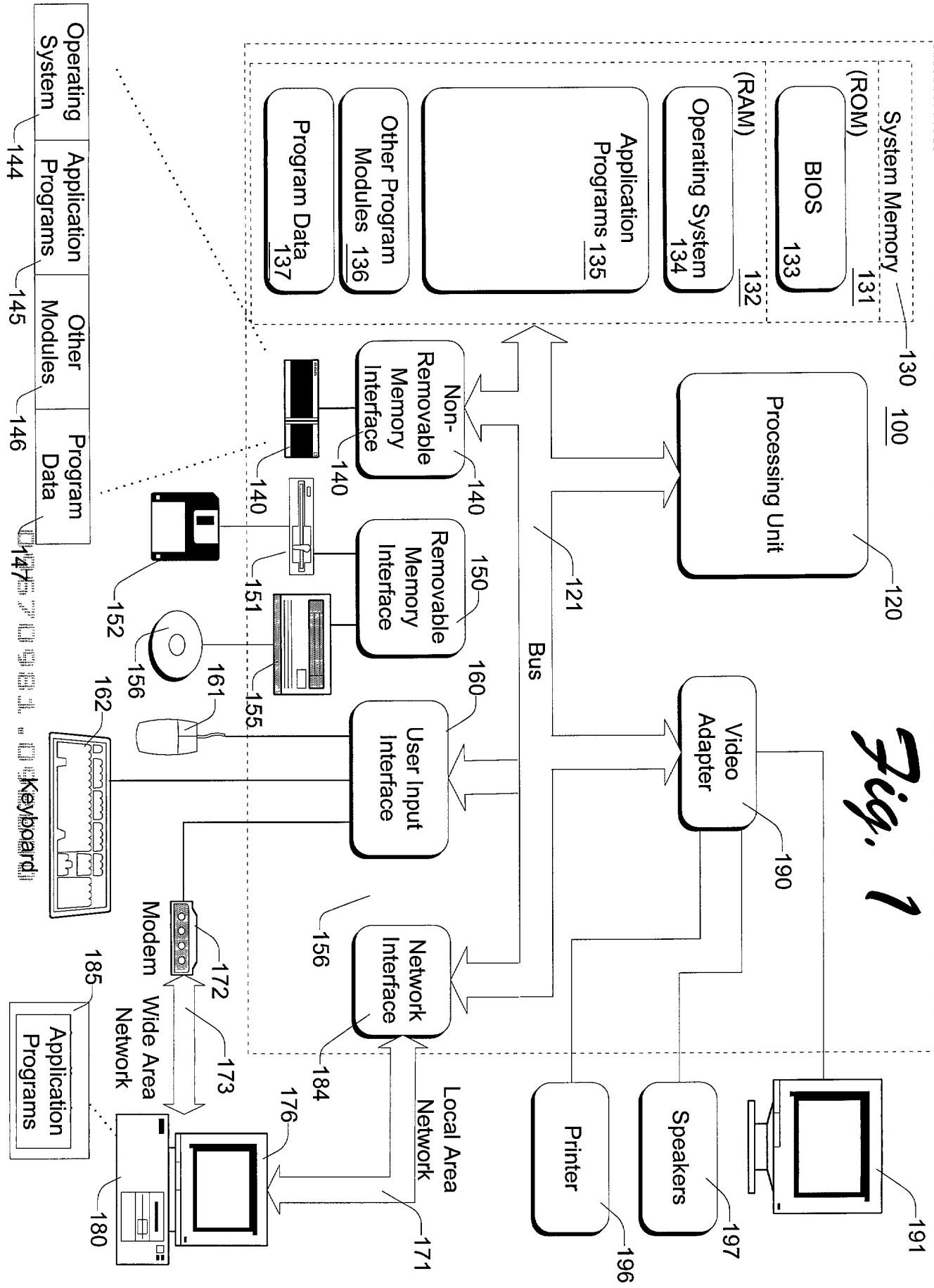
21
22 **51.** The communications protocol filter as recited in claim 45, wherein
23 the communications protocol is an Internet protocol and the communications
24 protocol data packets are Internet protocol data packets.

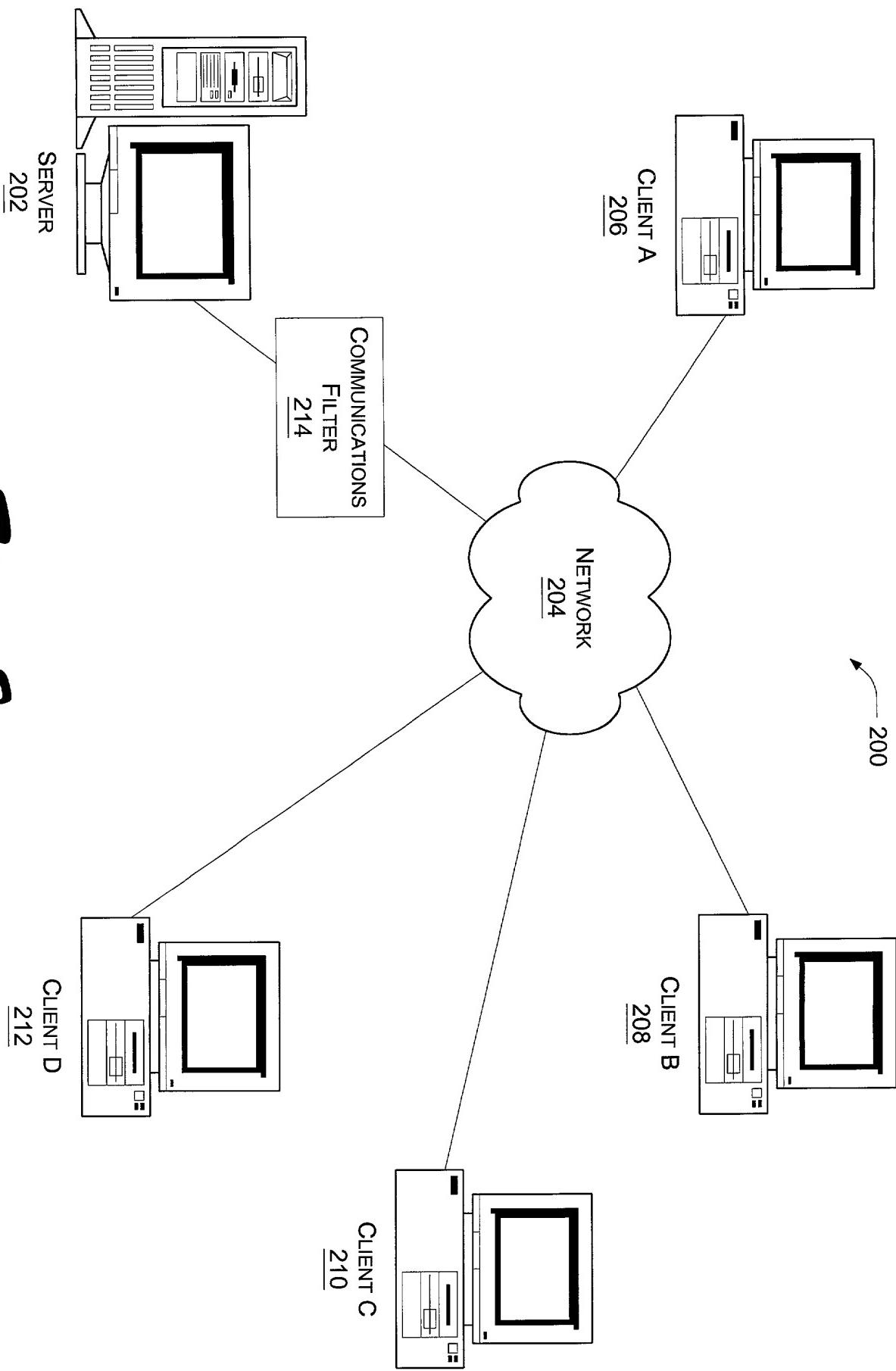
1 **ABSTRACT**

2 Systems and methods for controlling the number of clients that can access a
3 server in a server-client network are described. A communications filter driver
4 resides on the server and monitors network communications to determine a unique
5 network address of each client that attempts to access the server. A table of clients
6 that have accessed the server is stored at the server. When a client attempts to
7 access the server, the server allows the client to access the server if the network
8 address of the client is listed in the table. If the network address of the client is not
9 listed in the table, the server determines if a client limit has been met or exceeded.
10 If the limit has not been met, the client is allowed to access the server and the
11 Internet address of the client is added to the table.

12 If the limit has been reached, the server determines if the client is a client
13 that has previously accessed the server under a different network address. This is
14 accomplished by sending a signal to each network address listed in the table and
15 requesting a response. If a response is not received from a client, then the network
16 address for that client is removed from the table and the client attempting to access
17 the server is allowed to access the server. The network address for this client is
18 then added to the table. If a response is received from all network addresses, a
19 signal indicating that a client limit has been exceeded is sent to the client
20 attempting to access the server and access to the server is denied to that client.

Fig. 1





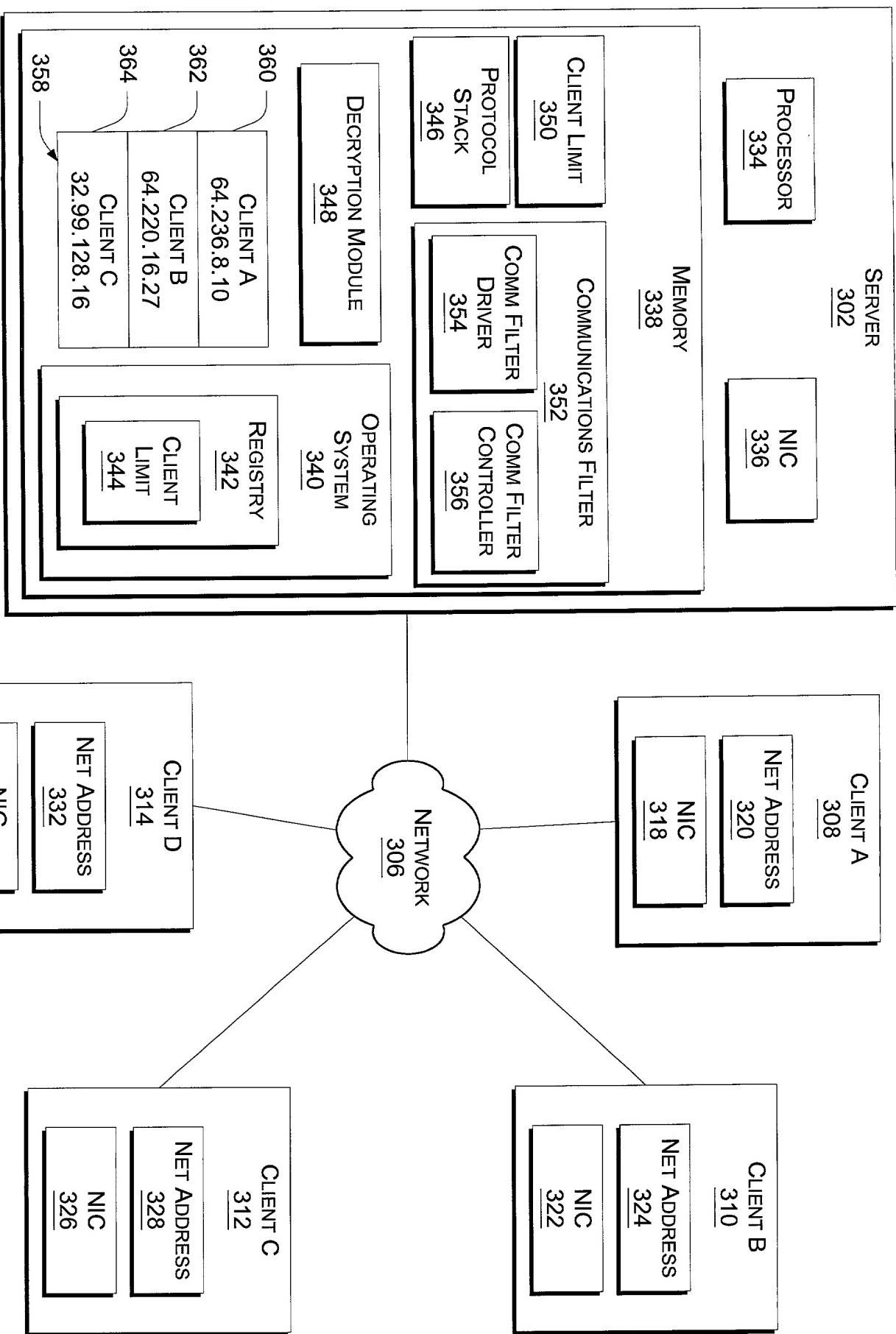


Fig. 3

